Deep Learning with CUDA
Dedicated architectures
Generative Adversarial Networks

Tomasz Szumlak

WFiIS AGH

03/04/2024, Kraków

AI going creative

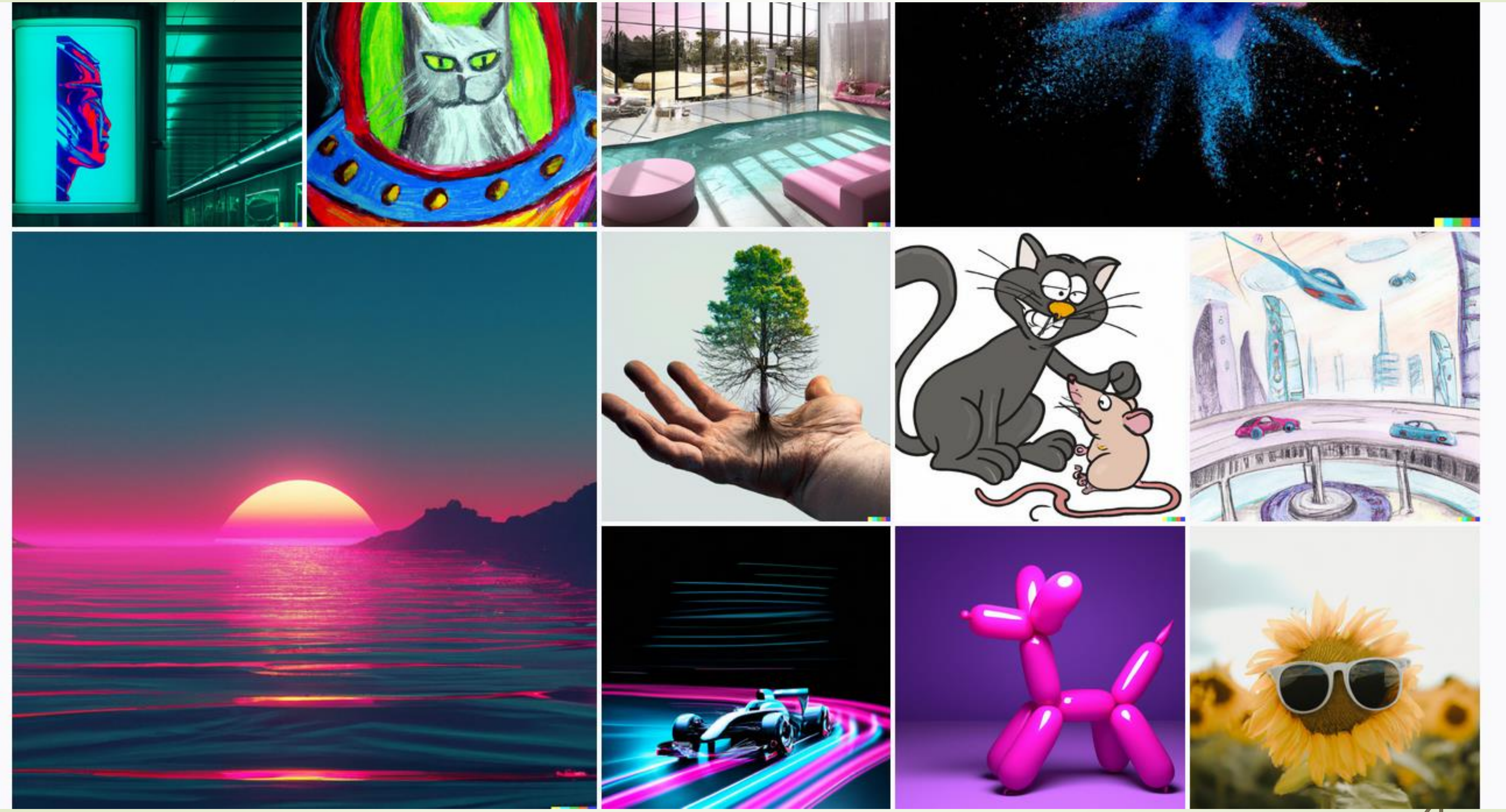
- It is possible to generate „new content" using noise (latent space), or data that are not correlated
- Two separate entities (ANNs) Generator and Discriminator
- Applications ranges from creating art, predicting new words, generating (augmenting) data to improve training
- It is believed at this point that our brains runs a very intricate generative model which is able to give us creativity (to see new things and sometimes to see things that are not there…)

- **GANs are without a doubt one of the most intriguing bits of ML**

AI going creative

arXiv: 1406.2661

From good fella from Open AI



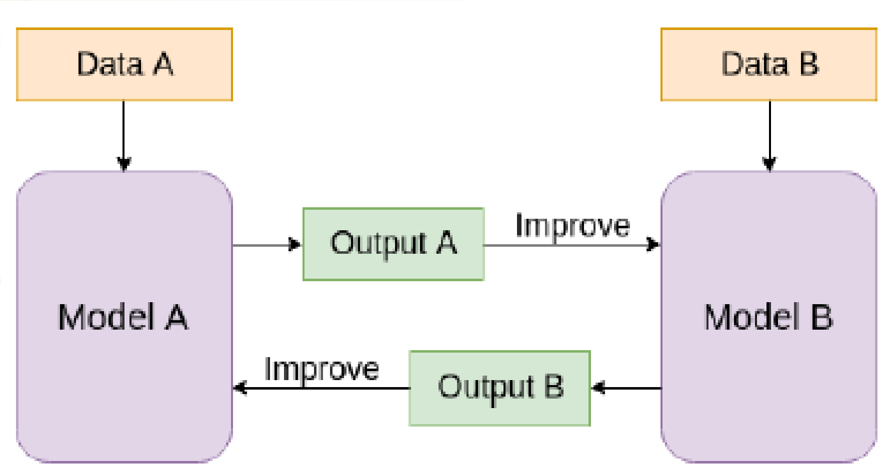
A tale of two kingdoms

- This story goes in different flavours, but the conclusions are always the same!
- Imagine you have two kingdoms, each have its own blacksmiths that can make armour and weapon. **One king never allowed any domestics contests** and the other **demanded constant cross-checks** of armour and weapon
- **You can guess which kingdom would be better in military technology!**
- **The same goes for the GAN approach – constant challenge**!

Event generators

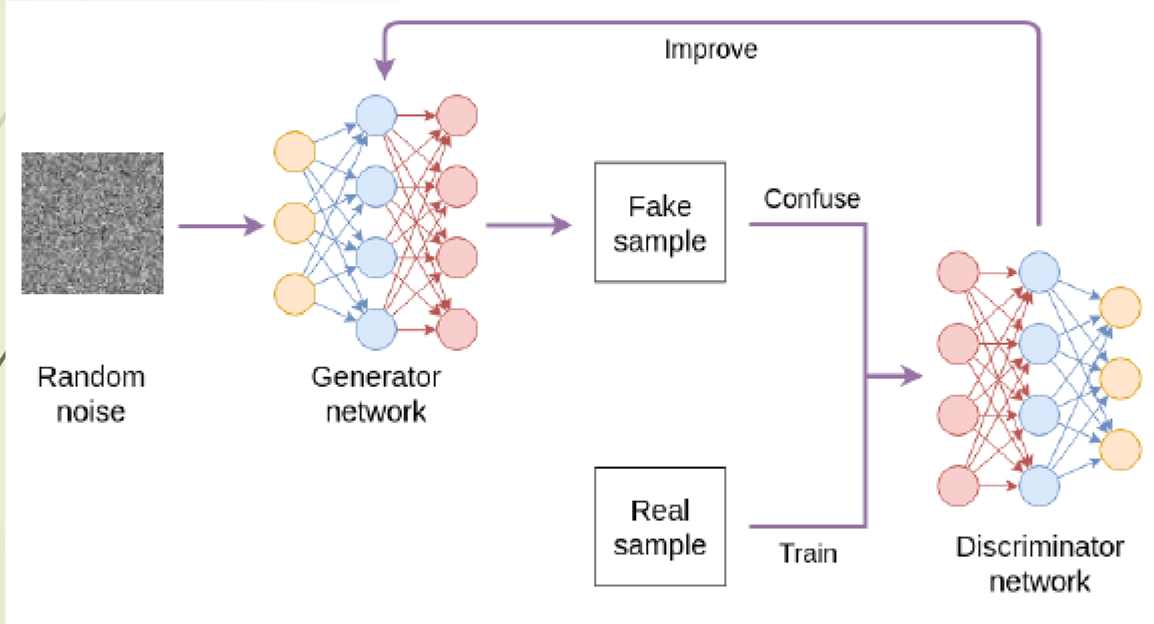
- The idea is actually quite old: physics generators that mimic Nature
- We can say that the **generators** tries to „**map low-dimension data to high-dimension data**"
- Classification models **do the opposite**!
- So, the training were two models make an attempt to weaken each other and on the long run enhance each other is called **adversarial learning**
- So, when designing a GAN system we need two models!

Adversarial systems



- Adversarial learning
  - Need two models A and B
  - The output of B is going to improve the A and the output of A do the same for B
  - One model will need „a real data" sample for training
  - Not all data are going to be fake

GAN architecture



- Here the generator model is using the noise to produce fake data that are fed to the second model
- The second one is a classification model that makes an attempt on detecting the fake data
- The differences between the generated and real data are used to improve the generator
- Real data are used to train the discriminator model

GAN optimisation rules

- Let set $\mathcal{G}$ and $\mathcal{D}$ to represent the generator and discriminator models respectively, the performance function is $\mathcal{V}$. The optimisation objective can be written as follow:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{V}(\mathcal{D},\mathcal{G}) = \mathbb{E}_{\vec{x}}[log\mathcal{D}(\vec{x})] + \mathbb{E}_{\vec{x}^*}\left[log\left(1-\mathcal{D}(\vec{x}^*)\right)\right]$$

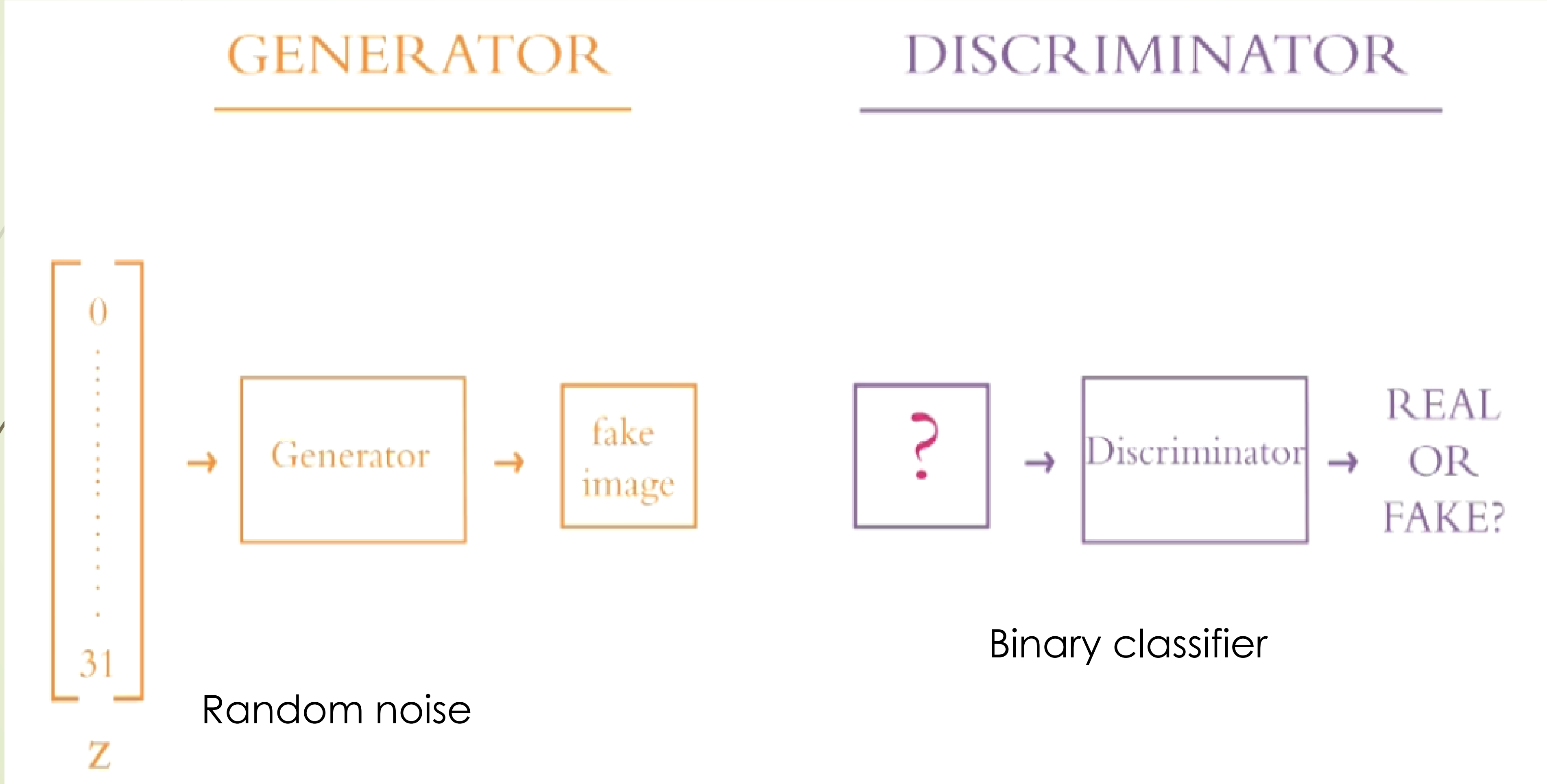
- Here: $\vec{x}$ - real samples, $\vec{x}^* = \mathcal{G}(z)$ - generated samples ("z" represents noise), $\mathbb{E}_{\vec{x}}[f]$ is the average value of any function over the sample space
- Model $\mathcal{D}$ should maximise the „good" prediction for the real sample - we are looking for the max – gradient ascent update rule

$$\vec{\theta}_{\mathcal{D}} \leftarrow \vec{\theta}_{\mathcal{D}} + r \cdot \frac{1}{m} \nabla_{\vec{\theta}_{\mathcal{D}}} \sum_{i/1}^{i/m} \left[log\mathcal{D}(\vec{x}) + log\left(1-\mathcal{D}(\vec{x}^*)\right)\right]$$

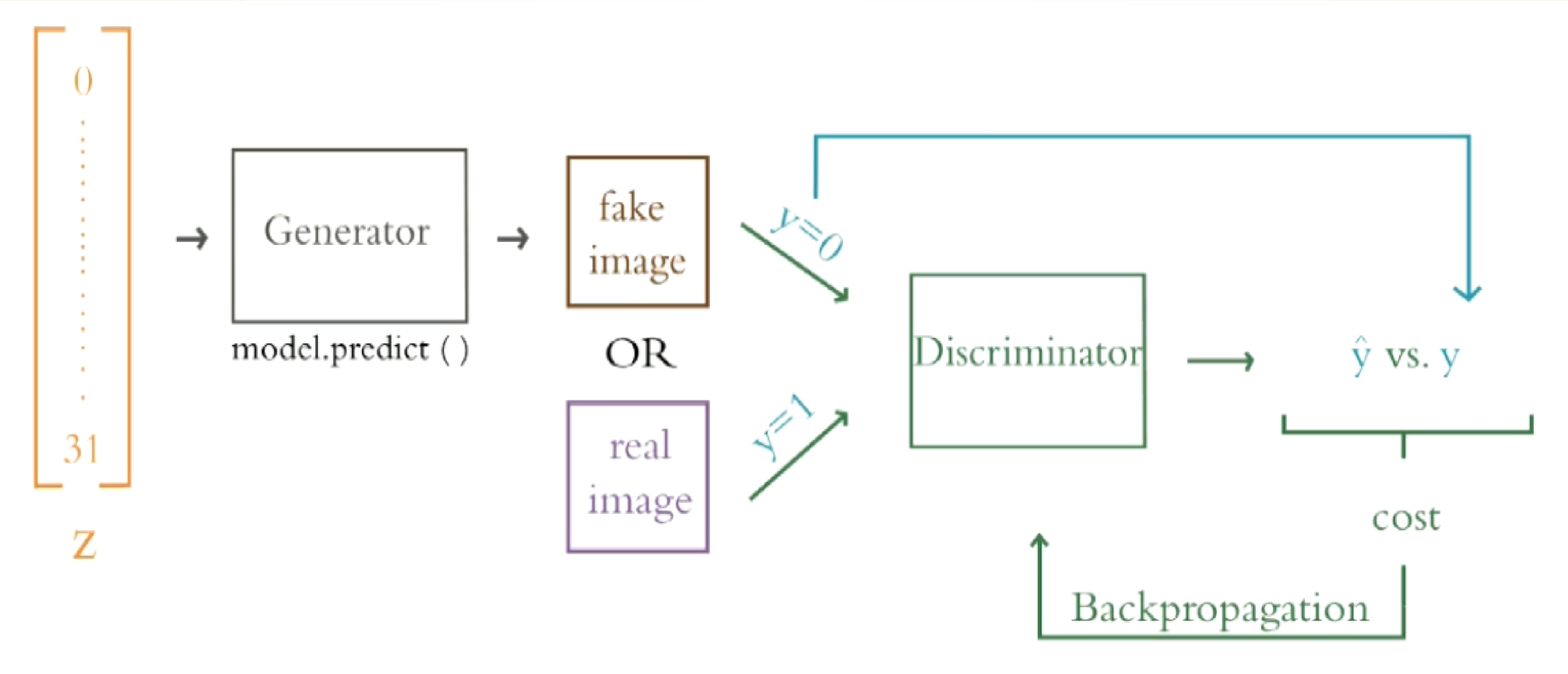
- Model $\mathcal{G}$ must trick the discriminator, thus, it minimises the $1-\mathcal{D}(\vec{x}^*) = 1-\mathcal{D}\left(\mathcal{G}(z)\right)$

$$\vec{\theta}_{\mathcal{G}} \leftarrow \vec{\theta}_{\mathcal{G}} - r \cdot \frac{1}{m} \nabla_{\vec{\theta}_{\mathcal{G}}} \sum_{i/1}^{i/m} \left[log\left(1-\mathcal{D}(\vec{x}^*)\right)\right]$$

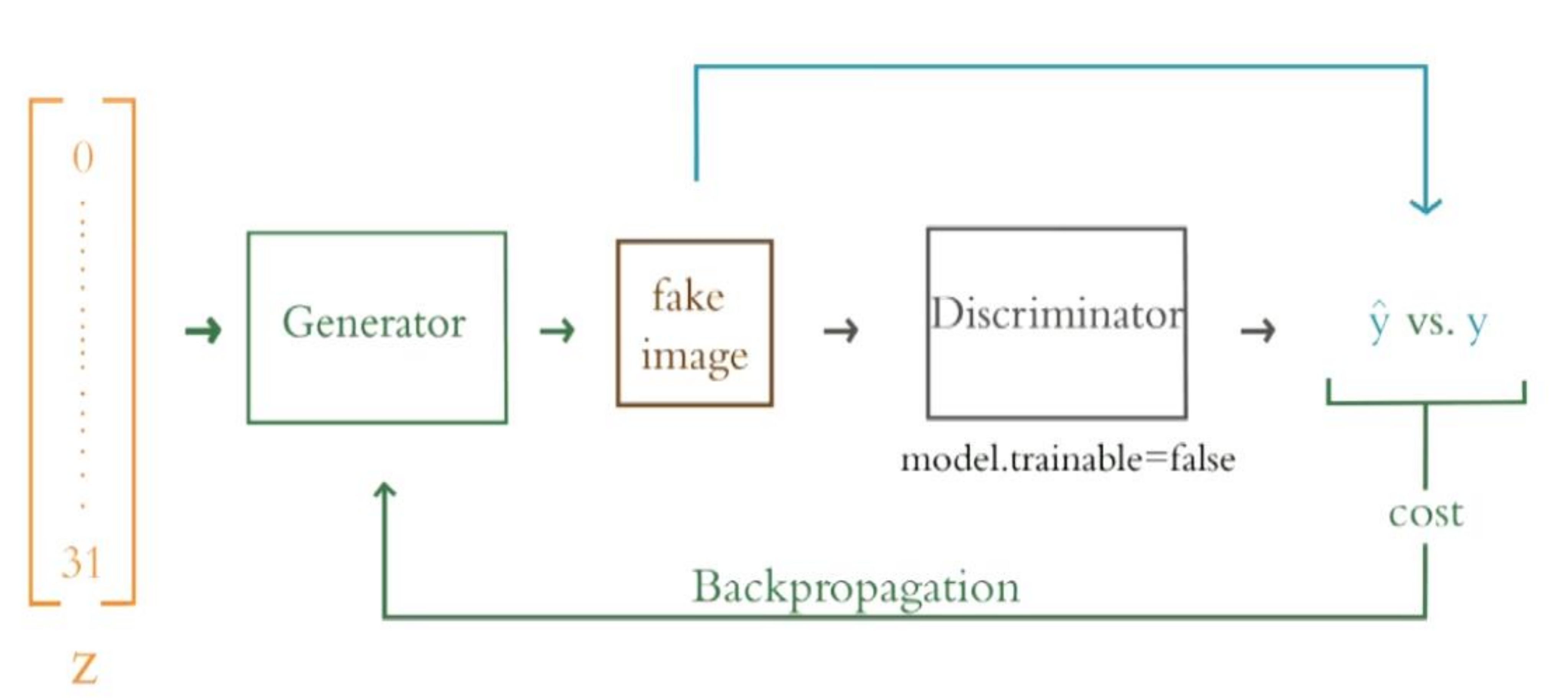
How it works?


GENERATOR
DISCRIMINATOR
0
31
Z
Generator
fake image
?
Discriminator
REAL OR FAKE?
Binary classifier
Random noise

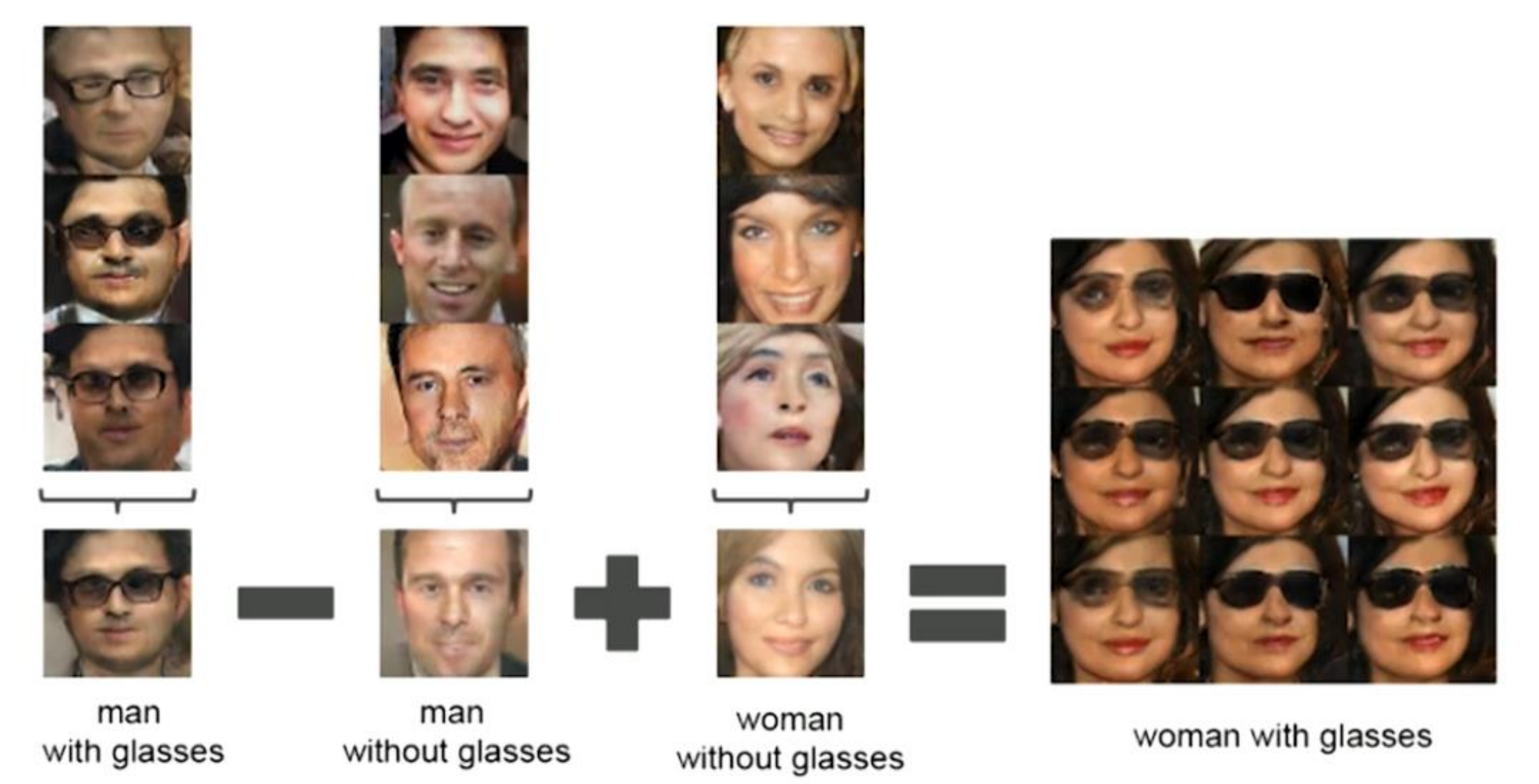
Discriminator training



Generator training

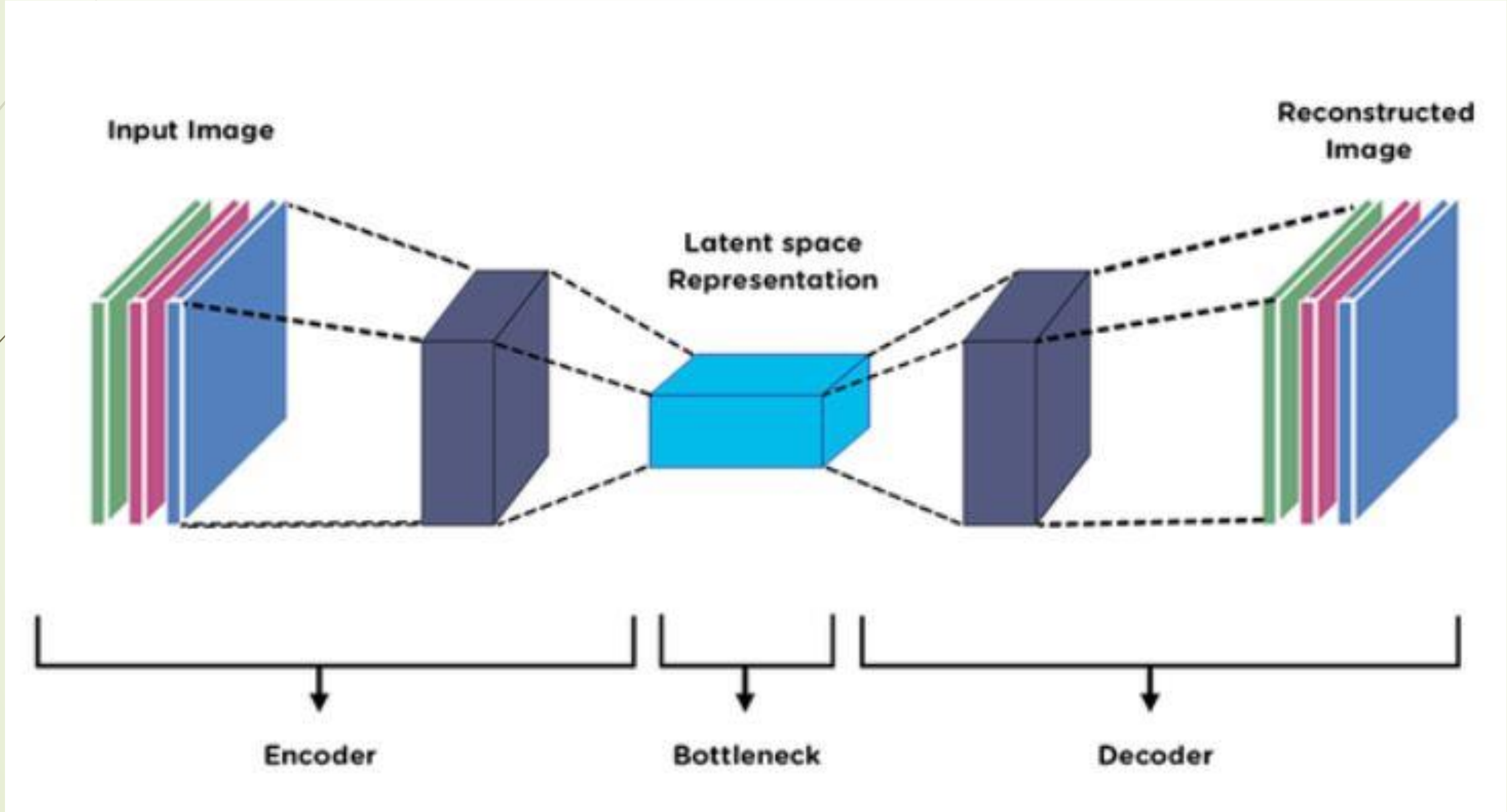


Latent space - DCGANs

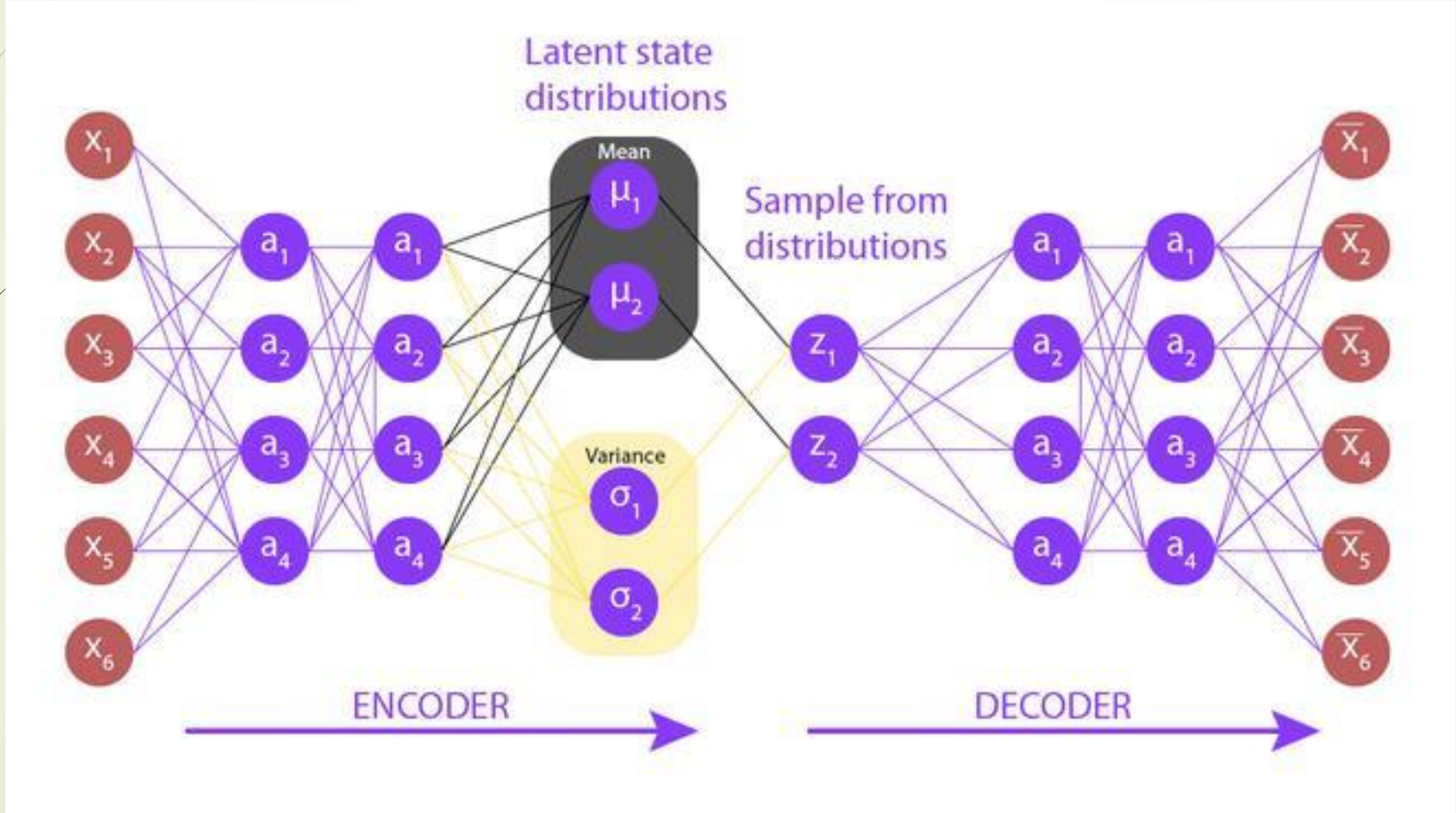


arXiv: 1511.06434

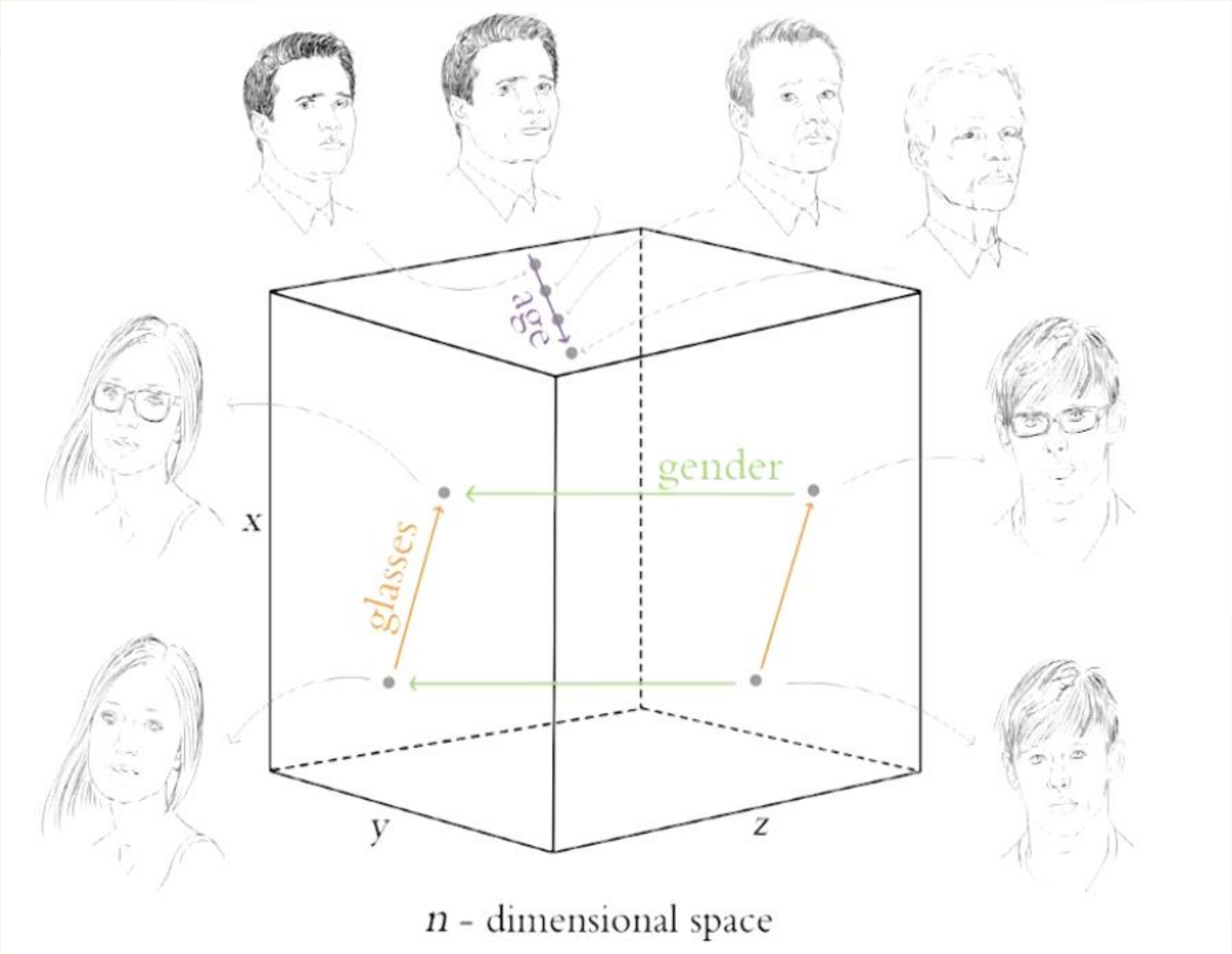
Latent space - Autoencoders



Latent space - Autoencoders



Latent space - DCGANs



https://www.youtube.com/watch?v=G06dEcZ-QTg&t=85s

The end